



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/699,074	10/30/2003	Riaz Merchant	64660(51919)	1373
21874	7590	05/04/2006	EXAMINER	
EDWARDS & ANGELL, LLP P.O. BOX 55874 BOSTON, MA 02205			MYINT, DENNIS Y	
			ART UNIT	PAPER NUMBER
			2162	

DATE MAILED: 05/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/699,074	Applicant(s) MERCHANT, RIAZ	
	Examiner Dennis Myint	Art Unit 2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 10/30/2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-59 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-59 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 30 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-59 have been examined.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 2 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 2, the claim is rejected under 35 U.S.C. 112, second paragraph, because the claim makes use of a Trademark "Microsoft" and "Windows" which belongs to Microsoft Corporation.

MPEP 2173.05(u) states that: "the presence of a Trademark or trade name in a claim is not, per se, improper that Trademarks or Trade Names in a Claim under 35 U.S.C. 112, second paragraph.....If the trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of the 35 U.S.C. 112, second paragraph. Ex parte Simpson, 218 USPQ 1020 (Bd. App. 1982)."

Any claim not specifically addressed is rejected by virtue of its dependency.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

Art Unit: 2162

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claim 1-43 and 46-58 are rejected under 35 U.S.C. 102(b) as being anticipated by Sierakowski et al., ("Migrating Btrieve Applications to MS SQL Server 7.0" in "SQL Server 7.0 Resource Guide", June 1999 <http://www.microsoft.com/technet/prodtechnol/sql/70/reskit/part11/sqc20.msp?mfr=true>).

As per claim 1, Sierakowski et al. is directed to a system (Sierakowski et al., Page 1, Chapter 21, "Migrating Btrieve Applications to MS SQL Server 7.0") for migrating an application developed around an ISAM database server to an SQL database server without source level changes ("Sierakowski et al., Page 5, "Step 1: Wrapper DLL" and "The goal of this stage in the conversion is to provide a layer of abstraction between the base application and Microsoft SQL Server. Using the concept of a wrapper DLL, the base application, Btrvapp.exe, can access SQL Server data without modification."), said system comprising:

a) a database migration tool (Sierakowski et al., Page 4 "Sample Application and Code Reference", "Btrvapp.exe", "Mybtrv32.dll", "Odbcapp.exe", and "Morepubs.sql");
and

b) a database driver (Sierakowski et al, Page 5, "a Pervasive Btrieve ODBC driver" and Page 4 "Mybtrv32.dll");

wherein said database migration tool migrates data from the ISAM database server to the SQL database server in such a manner that transparency of data operation is maintained from a perspective of the application ("Sierakowski et al., Page 5, "Step 1: Wrapper DLL" and "The goal of this stage in the conversion is to provide a layer of

abstraction between the base application and Microsoft SQL Server. Using the concept of a wrapper DLL, the base application, Btrvapp.exe, can access SQL Server data without modification.); and wherein said database driver intercepts functional calls (Sierakowski Page 13, "For example, in the case of this conversion strategy, a wrapper DLL can intercept Btrieve calls made to an application and ...") specifying any database operation made to the ISAM database server from the application and translates them into corresponding SQL functional calls and statements (Sierakowski et al., Page 4, "Mybtrv32.dll" and " Sample wrapper DLL that translates the Btrieve calls in Btrvapp.exe to ODBC and Transact-SQL calls.") in such a manner that allows complete transparency between the SQL database server and the application so as to allow the application to continue to perform as it normally does and continue to receive and send data to the SQL database server in a format it expects with the ISAM database server (Sierakowski et al., Page 5, "Step 1: Wrapper DLL" and "Using the concept of a wrapper DLL, the base application, Btrvapp.exe, can access SQL Server data without modification.").

As per claim 2, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool is a standard 32-bit application that runs on a MICROSOFT WINDOWS platform (Note: Microsoft Server 7.0 is a 32-bit application).

As per claim 3, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver uses a native low level API to communicate with the SQL database server (Sierakowski et al., Page 5, i.e. "Conversion Strategy" and "a full ODBC" and Page 11, i.e. Figure: "DTS Import Wizard" and "Destination: Microsoft SQL Server 7.0 Only [OLE DB Provider]").

As per claim 4, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver provides a direct connection to the SQL database server (Sierakowski et al., Page 5, i.e. "Conversion Strategy" and "a full ODBC" and Page 8, i.e. Figure: "DTS Import Wizard" and "Destination: Microsoft SQL Server 7.0 Only [OLE DB Provider]").

As per claim 5, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool sets up the SQL database server (Sierakowski et al., Page 8, i.e. "DTS Import Wizard").

As per claim 6, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool copies data at high speeds using native tools for fast data loading (Sierakowski et al., Page 5, i.e., "First, you create a data source name (DSN) by using an ODBC driver or an OLE DB provider.").

As per claim 7, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool copies data at high speeds using native high speed data loading mechanisms and application programming interfaces (Sierakowski et al., Page 6, i.e. "In the System DSN dialog box, click Add, and then configure the Btrieve data source, Make sure that the data source points to your database." and Figure: "Pervasive Software ODBC Interface for Windows").

As per claim 8, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool generates SQL scripts to create tables and indexes. (Sierakowski et al, Page 1, i.e. "Job Scheduling and Execution" and "Great flexibility is provided though a variety of scripting environments: Microsoft Visual Basic Scripting

Edition, Java scripting, Microsoft Windows NT commands and custom ODBC and OLE DB programs." and Page 13-14, i.e. "Creating the Wrapper DLL" and "Determining Which Functions to Wrap"). Generating SQL scripts to create tables and indexes are inherent in the migration feature of Microsoft SQL Server 7.0.

As per claim 9, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool is a GUI application that sets up a necessary environment and files that are later used by said database driver (Sierakowski et al, Page 7-9, i.e. Figure: "DTS Wizard", "DTS Import Wizard", and "Column Mappings and Transformation").

As per claim 10, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool translates database and security information from the ISAM database server to the SQL database server (Sierakowski et al, Page 2, i.e. "Security: Security administration is improved and simplified through better integration with Windows NT security and SQL Server roles. Windows NT integration includes authentication, support for multiple groups, grant/revoke/deny permission management activities and dynamic use of groups.");

wherein the database information comprises data files or table definitions; and
wherein the database information comprises index information (Sierakowski et al., Page 7, Figure: "DTS Wizard" and "Data Transformation Services Wizard: The Data Transformation Services makes it easy to import, export, validate and transform heterogeneous data using OLE DB and ODBC. This Wizard helps you perform the steps to import or export data between many popular data formats including

databases, spreadsheets, and text files.”). As such, not that, inherently, Microsoft SQL Server 7.0 translates all and any database and security information from the ISAM database (Btrieve) to the SQL database server.

As per claim 11, the claim is rejected on the same basis as claim 10. Note that Microsoft SQL Server 7.0 inherently performs all the features recited in claim 11. Sierakowski et al. is directed to the system as defined in claim 1,

Wherein said database migration tool reads table and index definitions (Sierakowski et al, “Page 9, Figure: “DTS Import Wizard” and “Select Source Tables”);

wherein said database migration tool performs data type translation by mapping data types from the ISAM database server to the SQL database server (It is inherent that Migration Wizard of Microsoft SQL Server 7.0 maps data types from those of the ISAM database to those of SQL Server. Microsoft SQL Server 7.0 employs “Pervasive Btrieve ODBC driver”, Sierakowski et al, Page 5);

wherein said database migration tool reads security information on files to be translated (This feature has been examined as per claim 10);

wherein said database migration tool generates migration reports and function call traces (Sierakowski et al.. Page 4, i.e. “Sample Application and Code reference” and “Btrvapp.exe: It is the starting point for the conversion strategy, and it is a simple data-entry and *reporting* application.”. Also note that it is well know in the art that migration/upgrade/update reports are generated any commercial database servers.);

wherein said database migration tool allows users to browse data before and after translation (Sierakowski et al, "Page 9-10, Figure: "DTS Import Wizard", "Column Mapping and Transformations" and "Select Your Tables");

wherein said database migration tool allows switching between the ISAM and the SQL database servers by just adding or removing driver name prefixes (Sierakowski et al., Page 7, i.e. Figure: "DTS Import Wizard" "Choose a Data Source"" and Page 8 Figure: "DTS Import Wizard" and "Choose A Destination". Note that by changing source and destination, respective database drivers are also changed and vice versa.);

wherein said database migration tool generates scripts for fast loading of data into native types by generating text files and scripts that can be used by the SQL database server for high-speed database migration (Sierakowski et al, Page 1, i.e. "Job Scheduling and Execution" and "Great flexibility is provided though a variety of scripting environments: Microsoft Visual Basic Scripting Edition, Java scripting, Microsoft Windows NT commands and custom ODBC and OLE DB programs." and Page 13-14, i.e. "Creating the Wrapper DLL" and "Determining Which Functions to Wrap");

wherein said database migration tool allows migrated tables to be removed or dropped from the SQL database server(Official note is taken that the concept of removing dropping tables during database is notoriously well known in the art.);

wherein said database migration tool allows data to be read back into a table of the ISAM database server from a corresponding migrated table of the SQL database server (It is inherent in Microsoft SQL Server 7.0 that translation/transformation can be done in either direction.); and

wherein said database migration tool generates auxiliary files with appropriate table information so as to allow said database driver to function properly in its task as functional translator (Use of auxiliary or temporary files are inherent in migration tools.).

As per claim 12, Sierakowski et al. is directed to the system as defined in claim 1, wherein a type of functional translation said database driver performs is dependent on the ISAM database server and the SQL database server between which said database driver acts as a middle-ware or bridge (Sierakowski et al, Page 5, "a Pervasive Btrieve ODBC driver" and said Pervasive Btrieve ODBC driver is a middleware.).

As per claim 13, official note is taken that the concept of finding, fetching records using indexes has been notoriously well known in the art since the inception of databases and database servers. See Page 16 of Sierakowski et al, i.e. "Understanding ODBC and SQL Implementation" and "Cursors and Btrieve files are accessed similarly, for example, through the use of FETCH, GET FIRST, and NEXT operations. Also see Page 27 of Sierakowski et al., especially on the part on "Recommendations for Creating Indexes" and "Database Index Guidelines".

Similarly, referring claims 14-16 official note is taken that updating records, saving updated records, deleting records and like functions are notoriously well know in the art.

As per claim 17, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver has an ability to start a transaction on the SQL database server and provide a same transactional functionality of the ISAM database server (Sierakowski et al., Page 17, i.e. "Alternatively, the wrapper DLL could retrieve data from SQL Server into buffers maintained on the client or another computer. The application then

Art Unit: 2162

fetches data from the buffers instead of from SQL Server directly by using ISAM-like processing techniques.”).

As per claim 18, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver (Sierakowski et al, Page 5, “a Pervasive Btrieve ODBC driver”) has an ability to send a transaction instruction to the SQL database server and make the transaction permanent by committing to disk (Sierakowski et al., Page 4, i.e. “Mybtrv32.dll” and “ Sample wrapper DLL that translates the Btrieve calls in Btrvapp.exe to ODBC and Transact-SQL calls.”). Note that Transact-SQL calls could commit transactions to disk.

As per claim 19, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver has an ability to issue an abort transaction command in an event of an error during a begin/end transaction block so as to allow the transaction to be rolled back restoring record buffers and tables to their original states (Sierakowski et al., i.e. “Handling Errors” and “The wrapper DLL must use Btrieve return codes when exiting each function. Each wrapper function must return B_NO_ERROR or a Btrieve error code corresponding to the type of error that was encountered. By using a valid Btrieve return code, the base application code does not know that its library function is accessing SQL Server instead of Btrieve. You must return the Btrieve return codes that are expected by the base application in order for the wrapper DLL to work properly.”). The feature of rolling back a transaction if said transaction is not successful are well known in the art.

As per claim 20, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver allows structure of an existing index to be modified by the

application (Sierakowski et al, Page 27, i.e., "Users of Microsoft SQL Server 7.0 can benefit from the new graphical SQL Query Analyzer and *Index Tuning Wizard*.").

Claims 21-24 are rejected on the same basis claim 20. Note Page 27 of Sierakowski et al, i.e. "Indexes to improve performance: You can optionally create unique or non-unique indexes on tables", and "Clustered Indexes". As such, it is inherent that indexes can be created or deleted or dropped.

As per claim 25, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver provides a mechanism to implement auto-increment fields that are available in many ISAM databases as well as SQL databases in such a way that the application sees no difference between the ISAM database server and the SQL database server even though the SQL database server handling is different. Sierakowski et al. does not elaborate on Pervasive Btrieve Driver. However, "Pervasive.SQL 2000 v7.82, Service Pack.2a Patch Update" (<http://www.pervasive.com/support/updates/Readsp2Server.htm>, August 2000) states that "AUTO-INCREMENT (Identity) " and "Change in behavior for implementing AUTO-INCREMENT (Identity) and "Old behavior: To implement the Auto-Increment column feature, you had to create an Identity column with a default of 0, then, as a separate step, create a unique index for each Identity Column. If a default value clause was not added when the column was created, then you have to insert 0 for each identity column in the row. " and "New behavior: The SQL Relational Database Engine now creates an index for an Identity (auto-increment) column. It will also default the value for the column to the value 0. The net result is that the value will increment to the next in the series for the column."

As per claim 26, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver provides support for case insensitive indexes available in most ISAM databases but likely absent in some SQL databases; and wherein said database driver provides support for an index that contains ascending and descending index segments in order to avoid costly ORDER BY clauses in an SQL statement. Note that Microsoft SQL Server 7.0 is case insensitive. In addition, Sierakowski et al. discloses index segments ("Clustered Index") stating "Clustered Indexes physically order the table data on the indexed columns." (Sierakowski et al., Page 27, "Recommendations for Crating Indexes")

As per claim 27, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver provides a mechanism to switch between record-at-a-time access provided by the ISAM database server and set-based access provided by the SQL database server so as to perform order entry or order update by using the record-at-a-time access while for reports or batch processes by using set-based access. Note that the migration/conversion system, which is integrated with Microsoft SQL Server 7.0, provides for implementing the Btrieve functions from within the wrapper (Sierakowski et al, Page 14, "Implementing the Btrieve Functions Within the Wrapper"). As such, it is inherent that Microsoft SQL Server 7.0 could inherently emulate any Btrieve functions, including the record-at-a-time access. Therefore, Microsoft SQL Server 7.0 could provided both record-at-time access and set-based access ("Clustered Indexes", Page 27). Also see Page 16 of Sierakowski et al.

As per claim 28, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver provides support for all authentication methods for the SQL database server since the ISAM database server may or may not provide a secure authentication based security service for accessing tables while this is a standard requirement in the SQL database server;

wherein said database driver automatically pops a login dialog box if a file is opened without being logged onto the SQL database server;

wherein said database driver provides support for a login command that can be added to an application either compiled in or externally to support security services;

wherein the login command creates a connection handle which uniquely identifies a user connection; wherein the connection handle is kept in memory in a data structure during execution of the application; and

wherein a logout command destroys the memory in the data structure and releases the connection handle. (Sierakowski et al, Page 2, i.e. "Security: Security administration is improved and simplified through better integration with Windows NT security and SQL Server roles. Windows NT integration includes authentication, support for multiple groups, grant/revoke/deny permission management activities and dynamic use of groups.") It is inherent that in Microsoft SQL Server 7.0 security features such as login commands, logout command, and their related functions are provided.

As per claim 29, Sierakowski et al. is directed to the system as defined in claim 1, wherein a file open command opens an auxiliary file that is needed to create a memory structure ("Btrieve posBlock Handle") about both ISAM and SQL tables since the ISAM

database server has to open a file before it can access a file whereas in the SQL database server there is no concept of file open;

wherein the auxiliary file contains information about table structure that is not supported by the SQL database server but is needed by the application;

wherein the auxiliary file is stored both as a binary and as a text file; and

wherein said database driver supports a close function by destroying all memory structure created by the file open command and closes a table handle for a table since closing a file of the ISAM database server means the handle for the file is released and the file is no longer used by the application whereas the SQL database server doesn't require a table to be closed. (Sierakowski et al., Page 15, i.e. "Addressing the Btrieve posBlock Handle"). Particularly note the disclosure that *"In the Btrieve environment, posBlock is a unique area of memory that is associated with each open file and that contains logical positional information to access records. The Btrieve libraries initialize and use this memory area to perform data functions. The Btrieve application inserts into every Btrieve call a pointer to the posBlock. The wrapper DLL does not need to maintain any Btrieve-specific data within the posBlock, so it is free to use this memory area for other operations. In the example DLL wrapper, the memory area is used to store the unique identifier for the SQL Server data affected by the requested operation. Regardless of the contents of the posBlock maintained by the wrapper DLL, each memory block must be unique to each corresponding SQL Server table set. For example, Btrvapp.exe references two Btrieve files, Sales.btr and Titlepub.btr, where Sales.btr contains sales information for each title and Titlepub.btr maintains the title and publisher for each title. These files correspond to*

the bsales and titlepublishers tables that were created in the pubs database by the sample script, Morepubs.sql. In Btrvapp.exe, the B_OPEN operation opens the requested Btrieve file and creates its corresponding posBlock. In the wrapper, the same posBlock now references a particular table by name. The wrapper DLL can be designed to store any form of a unique identifier that represents the SQL Server data that it accesses. Table names are used in the context of this migration strategy for ease of presentation. The keybuffer parameter contains the file name of the Btrieve file to be opened when B_OPEN is called. The wrapper DLL implementation of the B_OPEN function sets the posBlock equal to this file or table name."

As per claim 30, Sierakowski et al. directed to the system as defined in claim 1, wherein said database migration tool performs a convert database operation by creating a corresponding table in the SQL database server so as to form a newly created table and copying data of the ISAM database server to the newly created table (Sierakowski et al., Page 4, i.e. "Conversion Strategy", Page 5 Figure: "Conversion Strategy: Application Architecture Stages", and Page 9 Figure: "Column Mappings and Transformation", "Create Destination Table", and "Optionally, modify the CREATE TABLE statement that was automatically generated.").

As per claim 31, Sierakowski et al. is directed to the system as defined in claim 30, wherein said database migration tool during said convert database operation brings up another dialog box to allow the user to set migration options so as to form selections (Sierakowski et al., Page 9 Figure: "DTS Import Wizard" and "Select Your Tables.").

As per claim 31, Sierakowski et al. is directed to the system as defined in claim 31, wherein said selections are stored in auxiliary files called .INT (intermediate) file and .TD (table definition) file. Such auxiliary/intermediate files and table definition files are inherent in any database servers including Microsoft SQL Server 7.0.

As per claim 32, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database driver supports setting and fetching table and database attributes when requested by the application (Sierakowski et al., Page 9 Figure: "Column Mapping and Transformations" and Page 10 "You can use this functionality to help you check for potential year 2000 problems, and to change data to reflect standard coding such as country codes, state names, or phone number formatting.").

As per claim 34, official note is taken that the concept of using database attributes/table attributes such as number of records in a table, maximum allowable records, and file modes are notoriously well known in the art.

As per claim 35, Sierakowski et al. is directed to the system as defined in claim 33, wherein a table attribute is changing field names or field types ("You can use this functionality to help you check for potential year 2000 problems, and to change data to reflect standard coding such as country codes, state names, or phone number formatting.").

Claims 36 and 37 are rejected on the same basis as 27. Since the migration/conversion feature of Microsoft Server 7.0 integrates both functions of Btrieve and SQL Server, database driver of the said feature provides a mechanism to support additional commands specific to said database driver that result in increased performance,

such as commands which fetch a complete record (such as by Btrieve) or need columns (such as by a SQL Server).

As per claim 38, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool identifies any repository (Sierakowski et al., Page 9 Figure: "Select Your Source Tables") containing information regarding an ISAM database structure to migrate; wherein said database migration tool allows a user to choose which data files of the ISAM database server will be migrated; and wherein said database migration tool initiates migration (Sierakowski et al, Page 10 Figure: "Column Mapping and Transformations". Particularly note the Visual Basic Transformation Script in the window of the figure.).

As per claim 39, Sierakowski et al. is directed to the system as defined in claim 38, wherein the repository containing information regarding an ISAM database structure includes data dictionaries, file definitions, or file lists (Sierakowski et al., Page 8 Figure "Specify Table Copy or Query"). Note that in the Btrieve source tables, ISAM database structures such as data dictionaries, file definitions, or file lists are inherent.

Claim 40 is rejected on the same basis as claim 39.

As per claim 41, Sierakowski et al. is directed to the system as defined in claim 1, wherein said database migration tool allows a user to locate and select a repository containing information regarding an ISAM database structure (Sierakowski et al., Page 9 Figure: "Select Your Source Tables") and display file entries in a file list dialog box by virtue of said database migration tool working with the repository containing information

regarding an ISAM database structure (Sierakowski et al., Page 10 Figure: "Select Your Source Tables").

Claim 42 and 43 are rejected on the same basis as claim 39 and 40 respectively.

Claims 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, and 58 are rejected on the same basis as Claims 1, 9, 40, 39, 31, 31, 30, 8, 1, 31, 6, 7, and 32 respectively.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 44, 45, and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sierakowski et al. in view of Deleeuw (U.S. Patent Application Publication Number 2003/0115551).

Sierakowski et al. as applied to claim 1 above teaches saving the DTS packages the user could re-execute the migration/conversion routine (Sierakowski et al., Page 12 Figure: "SQL Server Enterprise Manager"). It is inherent that files that have been migrated to SQL Server are logged in the DTA package. However, Sierakowski et al. does not explicitly recite that prefix of a driver is added to the migrated file and said prefix is the file name of the driver dll. However, marking files that has been processed is well known in the

Art Unit: 2162

art. For example, Deleeuw teaches marking up files that have been processed (Deleeuw, Paragraph 0030).

Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to add the feature of making up files that have been processed, as taught by Deleeuw, to the system of Sierakowski et al. so that, in the resultant system, migrated files would be marked up either by a flag or adding a file name of respective driver dll file. One would have been motivated to do so because it is a well known practice in the art to mark up files that have been processed.

Claim 59 is rejected on the same basis claim 45.

Conclusion

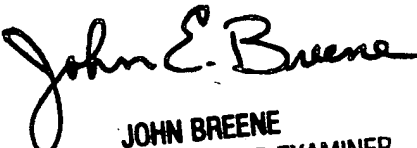
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis Myint whose telephone number is (571) 272-5629. The examiner can normally be reached on 8:30AM-5:30PM Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Dennis Myint

AU-2162


JOHN BREENE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100